International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 4, Issue 3, Jun 2014, 65-74
© TJPRC Pvt. Ltd.

CLUSTERING CONCEPTS USING FOR SEPARATED RANDOM DATA AND MATLAB

POOJA SHRIVASTAVA[1] & MANOJ SHUKLA[2]

[1]Research Scholar, Department of Computer Science and Engineering, Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India

[2]Associate Professor, Department of Computer Science and Engineering, Sunder Deep Group of Institution, Ghaziabad, Uttar Pradesh, India

ABSTRACT

Data mining is very vast area for research vast amount of data have been collected from biological science. Many researchers worked on in this area from different concepts and techniques. This paper presents the clustering concepts to solve the biological problem. In biological science protein sequences are complicated. So remove the complexity of dataset and create separated the random data. So in this paper uses the k-mean algorithm, MERPOS for protein data set, and MATLAB tool for protein analysis as data matrix form. Unsupervised learning is essentially a synonym for clustering.

KEYWORDS: K-Mean Algorithm, MATLAB, MEROPS, Unsupervised Learning

INTRODUCTION

Clustering is unsupervised learning. It's simple to use. Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups. Clustering solve the complexity problem to data set. Dissimilarities and similarities are based on describing the object. Similarity measure determine the similarity between two objects by the distance between them. This method apply the protein. Similarity measures play a fundamental role in the design of clustering method. In this paper presents the introduction of MEROPS online tool for protein data set, apply the k-mean algorithm used separated random protein data. And MATLAB use for data analysis. MATLAB is the technical computing. In this paper uses the MATLAB version R2007b.

OVERVIEW OF MEROPS (HTTP: // MEROPS. SANGER. AC. UK)

This database is a resource for catalogue and structure-based classification of peptidases. It contain a set of files, termed Pep Cards. Each of these cards provides information on a single peptidase. The information contains classification and nomenclature and gives hypertext links to the relevant entries in other database. The peptidases are classified into families on the basis Of statistically significant similarities between the protein sequences in the part termed the 'peptidase unit' which is most directly responsible for activity. In this tool suppose that I have taken 200 sample of protein as a object. And it's divided in to data matrix form. These data set use in MATLAB in data matrix form. MATLAB separated the random data of protein. These sample used as a Dataset, D, in data matrix form. It's the dataset of protein. And apply the k-means algorithm.

In below 100 data sample of protein define. For example:

Table 1

Name	Main Name	MEROPS Id
A-LAP	endoplasmic reticulum amino peptidase 1	M01.018
A20 peptidase	A20 peptidase	C64.003
A430081C19RIK (Mus musculus) protein	cytosolic carboxypeptidase 2	M14.029
A494 (*Arabidopsis thaliana*)	glycinain	C01.022
Aa2-001 protein (*Rattus norvegicus*)	Aa2-001 protein (*Rattus norvegicus*)	S60.974
AAA endopeptidase complex	AAA endopeptidase complex	XM41-001
AaaA aminopeptidase	AaaA aminopeptidase	M28.021
Aad peptidase	Aad peptidase	M15.012
AaH I	acutolysin A	M12.131
AaH III	acutolysin C	M12.303
AaHIV	fibrinogenase-2 (*Deinagkistrodon acutus*)	M12.334
AALP protein	aleurain	C01.041
Aap1' aminopeptidase	Aap1' aminopeptidase	M01.007
aarA-type peptidase	aarA-type peptidase	S54.004
AasP peptidase (*Actinobacillus pleuropneumoniae*)	AasP peptidase (*Actinobacillus pleuropneumoniae*)	S08.144
AaSPV1I (*Aedes aegypti*)	trypsin (mosquito type)	S01.130
agkistin (*Gloydius halys*)	jerdonitin	M12.313
AB5 subunit cytotoxin	cytotoxin SubAB	S08.121
ABC-protease	bacteriocin-processing peptidase	C39.001
AbgA putative peptidase	AbgA putative peptidase	M20.020
Abhd2a protein	Abhd2a protein	S33.A56
ABHD4 g.p. (*Homo sapiens*)	abhydrolase domain-containing protein 4	S33.013
Abhd8 g.p. (*Mus musculus*)	epoxide hydrolase-like putative peptidase	S33.011
abhydrolase domain-containing protein 4	abhydrolase domain-containing protein 4	S33.013
ABNORMAL LEAF SHAPE1 g.p. (*Arabidopsis thaliana*)	ALE1 peptidase	S08.014
AbpB g.p. (*Streptococcus gordonii*)	arginine aminopeptidase (*Streptococcus*-type)	C69.002
At3g61820 (*Arabidopsis thaliana*)	At3g61820 (*Arabidopsis thaliana*)	A01.A13

Table 1: Contd.,

AC-2 (*Haemonchus contortus*)	cathepsin B-like peptidase, nematode	C01.101
AC-2 g.p. (*Haemonchus contortus*)	papain homologue (nematode)	C01.055
AC-3 (*Haemonchus contortus*)	cathepsin B-like peptidase, nematode	C01.101
AC-4 (*Haemonchus contortus*)	cathepsin B-like peptidase, nematode	C01.101
AC-5 (*Haemonchus contortus*)	cathepsin B-like peptidase, nematode	C01.101
Ac-CP-1 (*Ancylostoma caninum*)	cathepsin B-like peptidase, nematode	C01.101
Ac-CP-2 (*Ancylostoma caninum*)	cathepsin B-like peptidase, nematode	C01.101
At4g00230 (*Arabidopsis thaliana*)	At4g00230 (*Arabidopsis thaliana*)	S08.A14
Ac1 proteinase	acutolysin A	M12.131
At4g04460 (*Arabidopsis thaliana*)	At4g04460 (*Arabidopsis thaliana*)	A01.A03
Ac2 proteinase	acutolysin A	M12.131
At4g10030 (*Arabidopsis thaliana*)	At4g10030 (*Arabidopsis thaliana*)	S33.A21
Ac3 proteinase (*Deinagkistrodon acutus*)	acutolysin C	M12.303
At4g16640 (*Arabidopsis thaliana*)	At4g16640 (*Arabidopsis thaliana*)	M10.A04
At4g16560 (*Arabidopsis thaliana*)-type peptidase	At4g16560 (*Arabidopsis thaliana*)-type peptidase	A01.A50
ACA(T)angiotensin-converting enzyme, testicularangiotensin-converting enzyme, C domain	angiotensin-converting enzyme peptidase unit 2	M02.004
Acasp-1 (*Ancylostoma caninum*)	necepsin-2	A01.068
ACC-C peptidase	ACC-C peptidase	S01.466
accessory gene regulator protein B (*Staphylococcus aureus*)	AgrB peptidase	C75.001
ACE	angiotensin-converting enzyme compound peptidase	XM02-001
Ace-MEP-6	MEP peptidase (nematode)	M13.011
At4g17486 g.p. (*Arabidopsis thaliana*)	At4g17486 g.p. (*Arabidopsis thaliana*)	C97.A05
ACE2	angiotensin-converting enzyme-2	M02.006
ACE3	Mername-AA152 protein	M02.971
ACEH	angiotensin-converting enzyme-2	M02.006

Table 1: Contd.,

Acer protein (*Drosophila melanogaster*)	peptidyl-dipeptidase Acer	M02.002
acetyl-lysine deacetylase	acetyl-lysine deacetylase	M20.975
Acetylcholinesterase	acetylcholinesterase	S09.979
acetylcholinesterase (*Caenorhabditis elegans*)	acetylcholinesterase (*Caenorhabditis elegans*)	S09.A89
acetylornithine deacetylase ArgE	acetylornithine deacetylase	M20.974
acetylornithine deacetylase form	yodQ g.p. (*Bacillus subtilis*)	M20.A21
acetylornithine deacetylase form	BSn5_04310 g.p. (*Bacillus subtilis*)	M20.A22
acetylornithine deacetylase form	PF1185 g.p. (*Pyrococcus furiosus*)	M20.A24
acey-1 (*Ancylostoma ceylanicum*)	cathepsin B-like peptidase, nematode	C01.101
AcfD (*Escherichia coli*)	YghJ g.p. (*Escherichia coli*)	M98.001
acg g.p. (*Aeromonas veronii*)	collagenase (*Salmonella*-type)	U32.003
Achelase	trypsin (invertebrate)	S01.112
Achelase	prothrombin activator (*Lonomia* sp.)	S01.420
Achromopeptidase	lysyl peptidase (bacteria)	S01.280
achromopeptidase component	beta-lytic metallopeptidase	M23.001
acid angiotensinase	lysosomal Pro-Xaa carboxypeptidase	S28.001
acid ceramidase precursor	acid ceramidase precursor	C89.001
acid endopeptidase, pepstatin-insensitive (*Bacillus*)	kumamolisin	S53.004
acid endopeptidase, pepstatin-insensitive (*Bacillus*)	kumamolisin-B	S53.005
acid extracellular protease (*Yarrowia lipolytica*)	axp peptidase (*Yarrowia lipolytica*)	A01.036
acid metalloendopeptidase, fungal	penicillolysin	M35.001
acid metalloproteinase	deuterolysin	M35.002
acid peptidase (*Cladosporium*)	acid peptidase (*Cladosporium*)	A9G.017
acid peptidase (*Paecilomyces*)	acid peptidase (*Paecilomyces*)	A9G.018
acid prolyl endopeptidase (*Aspergillus* sp.)	acid prolyl endopeptidase (*Aspergillus* sp.)	S28.004
acid protease (*Cladosporium*)	acid peptidase (*Cladosporium*)	A9G.017
acid protease (*Paecilomyces*)	acid peptidase (*Paecilomyces*)	A9G.018

Table 1: Contd.,

acidic dipeptidase	glutamate carboxypeptidase II	M28.010
acidic haemorrhagin	acutolysin A	M12.131
ACLF	venom metallopeptidase PREH	M12.160
ACLH	venom metallopeptidase PREH	M12.160
ACLHT	venom metallopeptidase PREH	M12.160
ACLP	adipocyte-enhancer binding protein 1	M14.951
ACLPREF	venom metallopeptidase PREH	M12.160
ACLPREH endopeptidase (*Agkistrodon contortrix laticinctus*)	venom metallopeptidase PREH	M12.160
AcNPV protease	V-cath peptidase	C01.083
ACO protein	kallikrein-related peptidase 15	S01.081
At4g17890 (*Arabidopsis thaliana*)	At4g17890 (*Arabidopsis thaliana*)	C19.A16
acp1 g.p. (*Sclerotinia sclerotiorum*)	peptidase EapC	G01.004
At4g20070 (*Arabidopsis thaliana*)	At4g20070 (*Arabidopsis thaliana*)	M20.A07
ACP2 *Entamoeba histolytica*	histolysain	C01.050
Acrocylindropepsin	acrocylindropepsin	A9G.019
Acrolysin	acrolysin	M9G.003
Acrosin	acrosin	S01.223
ADAM28 peptidase (mouse-type)	ADAM28 peptidase (mouse-type)	M12.020
ADAM28 peptidase (*Homo sapiens*-type)	ADAM28 peptidase (*Homo sapiens*-type)	M12.224
ADAM29 protein	ADAM29 protein	M12.981

And protein create as a object in this experiment. As so on 200 data sample we use.

Introduction of k-Mean Algorithm

IDX = k means(X, k) partitions the points in the n-by-p data matrix X into k clusters. This iterative partitioning minimizes the sum, over all clusters, of the within-cluster sums of point-to-cluster-centroid distances. Rows of X correspond to points, columns correspond to variables. K means returns an n-by-1 vector IDX containing the cluster indices of each point. By default, k means uses squared Euclidean distances. When X is a vector, k means treats it as an *n*-by-1 data matrix, regardless of its orientation.

[IDX, C] = k means(X, k) returns the k cluster centroid locations in the k-by-p matrix C.

[IDX, C, sum d] = k means(X, k) returns the within-cluster sums of point-to-centroid distances in the 1-by-k vector sum d.

[IDX, C, sum d, D] = k means(X, k) returns distances from each point to every centroid in the n-by-k matrix D.

[...]=k means (...,*param1*, *val1*, *param2*, *val2*,...) enables you to specify optional parameter/value pairs to control the iterative algorithm used by k means. Valid parameter strings are listed in the following table.

Syntax

IDX = k means(X, k) [IDX, C] = k means(X, k) [IDX, C, sum d] = k means(X, k) [IDX, C, sum d, D] = k means(X, k) [...] = k means(...,*param1*,*val1*,*param2*,*val2*,...)

About MATLAB

I uses the MATLAB version R2007b. **MATLAB** (**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran. Last release for Windows 2000 and Power PC Mac. License Server support for Windows Visita-New internal format for P-code.

IMPLEMENTATION AND RESULTS

Main objective is in this implementation is maintain the distance and separated the random data of protein dataset and remove the complexity of dataset. In this observation protein data set as a matrix form and implement on the matlab. Firstly for implementation uses the MATLAB. Select k observations from X at random (default). Second Select k points uniformly at random from the range of X. Not valid with Hamming distance. Select the dataset D from MEROPS online tool. These dataset is protein as a matrix form. Perform a preliminary clustering phase on a random 10% subsample of X. This preliminary phase is itself initialized using 'sample'. X is a data matrix. K - by-p matrix of centroid starting locations. In this case, you can pass in [] for k, and k means infers k from the first dimension of the matrix. You can also supply a 3-D array, implying a value for the 'replicates' parameter from the array's third dimension.

```
X = [randn (100,2)+ones(100,2);...

randn(100,2)-ones(100,2)];

opts = statset('Display','final');

[idx,ctrs] = kmeans(X,2,...

'Distance','city',...

'Replicates',5,...

'Options',opts);
```

Where X is the data matrix. In this implementation two cluster are used. 100 random data sample in one cluster. And other 100 random data sample is other second cluster. So total is 200 sample as matrix form from protein dataset D. after that create the total sum of distances.

In protein dataset, for two clusters, specify five replicates, and use the 'display' parameter to print out the final sum of distances for each of the solutions.

5 iterations, total sum of distances = 284.671

4 iterations, total sum of distances = 284.671

4 iterations, total sum of distances = 284.671

3 iterations, total sum of distances = 284.671

3 iterations, total sum of distances = 284.671

Overall 5 iterations, total sum of distances = 284.671

And after that to plot a graph of separated random data.

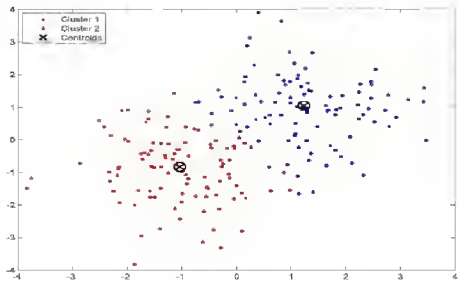

Figure 1: Graph of Separated Random Data

Above this figure we can see that how data is separated.

Experiment Result

After implementation found the distances min and max form. We can see that what is the value of initial data sample X. and after that what is the value of output between the dataset of protein it's maintain the distance in between the data set and increases the value. Found the gap here in between the data sample. Uses the two clusters in this experiment. In below the table we can see that value of separated data. 10% subsample also given the value of min and max.

Table 2: K-Mean Algorithm Output Table

Name	Intra-Cluster Distances (Min)	Inter-Cluster Distances (Max)
Data Sample(X) Initial	-3.44	3.1832
Ans (output value)	-3.37	4.2025
Sub Sample(x)	-1.17	1.1139

Table 3: Show that Iteration and Total Sum of Distances

Iteration	Total Sum of Distances
5	284.671
4	284.671
4	284.671
3	284.671
3	284.671

In above table we can see that value of initial data set and subsample of data set D.

plot(X(idx==1,1),X(idx==1,2),'r.','MarkerSize',12)

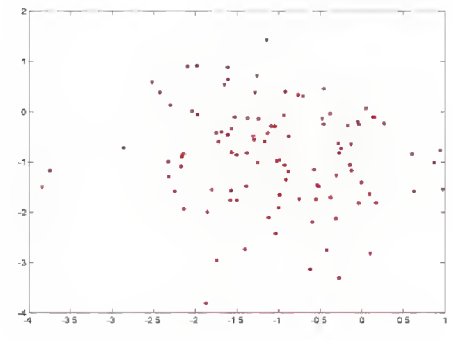

Figure 2: Graph of Cluster1

plot(X(idx==2,1),X(idx==2,2),'b.','MarkerSize',12)

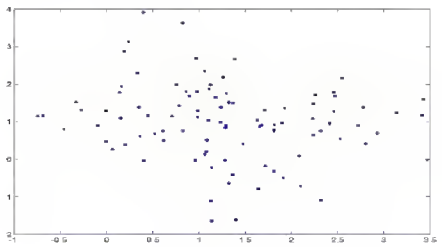

Figure 3: Graph of Cluster2

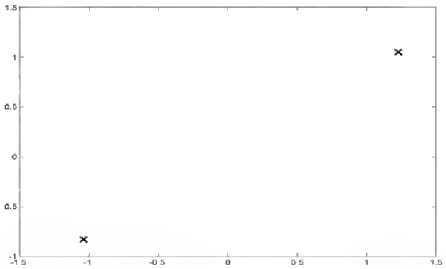

Figure 4: Centroids of the Points1

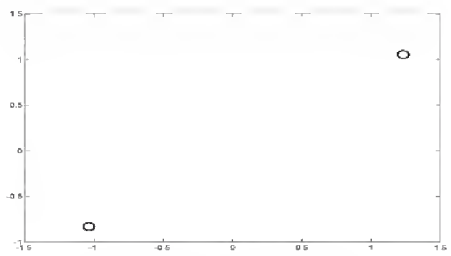

Figure 5: Centroids of Point 2

Above these figure we can see that how the cluster is working. And final graph is the data set is separated.

CONCLUSIONS

The main objective in this paper how data mining use in any field. And in biological science protein analysis are complicated. So in this paper with the help of k-mean algorithm and MATLAB tool we remove the complexity and found

the accuracy and increase the gape in between the protein data set. Also define the how the data is separated and how clustering is working.

REFERENCES

1. Data mining concepts and techniques (Jiawei han, Micheline Kamber,Jian Pei).
2. G. Chamundeswari, Prof. G. Pardasaradhi Varma, Prof. Ch. Satyanarayana International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 5, July - 2012 ISSN: 2278-0181.
3. http:// merops. sanger.ac.uk.
4. Sudip Poddar & Anirban Mukhopadhayay.
5. Introduction to Data Mining by Tan, Steinbach, Kumar.
6. www.google.com